

WHAT IS CLAIMED IS:

- 1 1. A database engine comprising:
2 a transactional mechanism supporting heterogeneous distributed transactions, said
3 transactional mechanism having
4 means for recognizing data sources conforming to the X/Open XA standards, said
5 data sources including structured and non-structured external data sources; and
6 means for managing transactions in which said data sources participate.
- 1 2. The database engine according to claim 1, in which said transactional mechanism
2 further comprises, for each of said data sources:
3 means for supporting transactional events conforming to the X/Open XA standards,
4 said transactional events including prepare, commit, rollback, redo, and undo.
- 1 3. The database engine according to claim 1, further comprises:
2 support functions configured to support each recognized data source.
- 1 4. The database engine according to claim 3, further comprises:
2 means for invoking said support functions at appropriate transactional events, said
3 transactional events including prepare, commit, rollback, redo, and undo.
- 1 5. The database engine according to claim 1, in which
2 said database engine supports at least one database application; wherein
3 each of said data sources has one or more instances; and wherein
4 said at least one database application interacts with said one or more instances via
5 said database engine.
- 1 6. The database engine according to claim 1, wherein
2 each of said data sources is a resource manager assigned with a unique identifier.

- 1 7. The database engine according to claim 1, wherein said transactional mechanism
2 further comprises:
3 means for generating and maintaining a global transaction ID for each of said
4 heterogeneous distributed transactions; and
5 means for producing a 2-phase commit transaction model for said data sources.
- 1 8. A computer system implementing the database engine of claim 1, wherein said
2 computer system is programmed to:
3 support said heterogeneous distributed transactions accessing said data sources
4 including said structured and non-structured external data sources;
5 recognize said data sources; and
6 manage said transactions in which said data sources participate.
- 1 9. A computer readable medium storing a computer program implementing the database
2 engine of claim 1, said computer program comprising computer-executable instructions for:
3 recognizing said data sources;
4 assigning each of said data sources with a unique identifier;
5 generating one or more instances for each of said data sources;
6 configuring support functions to support said data sources;
7 managing said transactions in which said data sources participate;
8 generating and maintaining a global transaction ID for each of said heterogeneous
9 distributed transactions;
10 invoking said support functions at appropriate transactional events including begin,
11 prepare, commit, rollback, redo, and undo; and
12 producing a 2-phase commit transaction model for said data sources.
- 1 10. A database server comprising:
2 a database engine comprising
3 a transactional mechanism supporting heterogeneous distributed transactions,
4 said transactional mechanism having

5 means for recognizing data sources conforming to the X/Open XA standards,
6 said data sources including structured and non-structured external data sources;
7 support functions configured to support each recognized data source;
8 means for managing transactions in which said data sources participate; and
9 means for invoking said support functions at appropriate transaction events
10 including prepare, commit, and rollback.

1 11. The database server according to claim 10, further comprising:
2 at least one database application; wherein
3 said database engine supports said at least one database application; wherein
4 each of said data sources has one or more instances; and wherein
5 said at least one database application interacts with said one or more instances via
6 said database engine.

1 12. The database server according to claim 11, wherein
2 each of said data sources is a resource manager assigned with a unique identifier.

1 13. The database server according to claim 10, further comprising:
2 means for generating and maintaining a global transaction ID for each of said
3 heterogeneous distributed transactions; and
4 means for producing a 2-phase commit transaction model for said data sources.

1 14. A computer system implementing the database server of claim 10, wherein said
2 computer system is programmed to:
3 support said heterogeneous distributed transactions accessing said data sources
4 including said structured and non-structured external data sources;
5 recognize said data sources; and
6 manage said transactions in which said data sources participate.

1 15. A computer readable medium storing a computer program implementing the database
2 server of claim 10, said computer program comprising computer-executable instructions for:
3 recognizing said data sources;
4 assigning each of said data sources with a unique identifier;
5 generating one or more instances for each of said data sources;
6 configuring support functions to support said data sources;
7 managing said transactions in which said data sources participate;
8 generating and maintaining a global transaction ID for each of said heterogeneous
9 distributed transactions;
10 invoking said support functions at appropriate transactional events including begin,
11 prepare, commit, rollback, redo, and undo; and
12 producing a 2-phase commit transaction model for said data sources.

1 16. A method of integrating a database system to support heterogeneous distributed
2 transactions, comprising:
3 recognizing data sources conforming to the X/Open XA standards, said data sources
4 including structured and non-structured data sources external to said database system; and
5 configuring a database engine with a transactional mechanism, said transactional
6 mechanism managing said heterogeneous distributed transactions in which said data sources
7 participate, wherein said transactional mechanism is capable of
8 assigning each of said data sources with a unique identifier;
9 generating one or more instances for each of said data sources;
10 generating and maintaining a global transaction ID for each of said
11 heterogeneous distributed transactions;
12 invoking support functions for said data sources at appropriate transactional
13 events; and
14 producing a 2-phase commit transaction model supporting said heterogeneous
15 distributed transactions with said data sources.

- 1 17. The method according to claim 16, further comprising:
2 constructing support functions for each of said data sources that participates in said
3 heterogeneous distributed transactions.
- 1 18. The method according to claim 16, wherein
2 said transactional events conform to the X/Open XA standards; and wherein
3 said transactional events include begin, prepare, commit, rollback, redo, and undo.
- 1 19. A computer system programmed to implement the method as set forth in claim 16,
2 including implementing support functions for each of said data sources that participates in
3 said heterogeneous distributed transactions; wherein
4 said transactional events conform to the X/Open XA standards; and wherein
5 said transactional events include begin, prepare, commit, rollback, redo, and undo.
- 1 20. A computer readable medium storing a computer program implementing the method
2 as set forth in claim 16, said computer program further implementing support functions
3 support functions for each of said data sources that participates in said heterogeneous
4 distributed transactions; wherein
5 said transactional events conform to the X/Open XA standards; and wherein
6 said transactional events include begin, prepare, commit, rollback, redo, and undo.